

IN THE CLAIMS:

Please amend the claims as indicated below.

1. (Currently Amended) A method for analyzing behavior of a software
5 system, comprising:

collecting details associated with a program task associated with said
software system based on a specification associated with said program task, wherein said
specification contains one or more conditions to initiate a trace of said program task; and
providing said collected details for analysis.

10 2. (Currently Amended) The method of claim 1, wherein a duration of said
program task is defined by said one or more conditions associated with a state of said
software system.

15 3. (Original) The method of claim 2, wherein said one or more conditions
includes an entry or exit of at least one specified method.

4. (Original) The method of claim 2, wherein said one or more conditions
includes a creation or deletion of at least one specified object.

20 5. (Original) The method of claim 2, wherein said one or more conditions
includes an invocation of at least one specified object.

6. (Original) The method of claim 2, wherein said one or more conditions
25 includes a passing of at least one specified object or scalar value as an argument, return
value or field value.

7. (Original) The method of claim 2, wherein said one or more conditions
includes at least one specified sequence of method invocations.

8. (Original) The method of claim 2, wherein said one or more conditions includes at least one specified resource exceeding at least one specified threshold.

9. (Original) The method of claim 1, wherein said collected details include
5 an existence or sequence of specified method invocations.

10. (Original) The method of claim 1, wherein said collected details include an existence or sequence of specified object creations and deletions.

10 11. (Original) The method of claim 1, wherein said collected details include an existence or sequence of specified class loading and unloading.

12. (Original) The method of claim 1, wherein said collected details include values of specified arguments to invocations of specified methods.

15

13. (Original) The method of claim 1, wherein said collected details include values of specified return values from invocations of specified methods.

14. (Original) The method of claim 1, wherein said collected details include
20 values of specified field values for invoked objects or field values for passed arguments.

15. (Original) The method of claim 1, further comprising the step of collecting said details for at least one specified number of task instances.

25 16. (Original) The method of claim 1, further comprising the step of collecting said details for at least one specified number of threads.

17. (Currently Amended) The method of claim 1, further comprising the step of dynamically modifying said specification associated with said program task
30 ~~specification~~ associated with said analysis in an iterative process.

18. (Currently Amended) The method of claim 1, further comprising the step of dynamically modifying a said specification ~~of~~ to identify which details to collect in an iterative process.

5 19. (Original) The method of claim 1, further comprising the step of connecting to a running version of said software system.

20. (Original) The method of claim 1, further comprising the step of visually analyzing said collected details.

10

21. (Original) The method of claim 1, further comprising the step of visually analyzing said collected details for a plurality of instances of said program task.

22. (Original) The method of claim 1, further comprising the step of
15 quantitatively analyzing said collected details.

23. (Original) The method of claim 1, further comprising the step of quantitatively analyzing said collected details for a plurality of instances of said program task.

20

24. (Currently Amended) A method for tracing details associated with a program task executing in a software system, comprising:

monitoring said software system to identify said program task based on a specification associated with said program task, wherein said specification contains one
25 or more conditions to initiate a trace of said program task; and
providing trace ~~tracing~~ details associated with said program task.

25. (Currently Amended) The method of claim 24, wherein a duration of said program task is defined by said one or more conditions associated with a state of said
30 software system.

26. (Original) The method of claim 25, wherein said one or more conditions is selected from the group consisting essentially of (i) an entry or exit of at least one specified method, (ii) a creation or deletion of at least one specified object, (iii) an invocation of at least one specified object, (iv) a passing of at least one specified object or scalar value as an argument, return value or field value, (v) at least one specified sequence of method invocations, and (vi) at least one specified resource exceeding at least one specified threshold.

27. (Original) The method of claim 24, wherein said collected details include at least one of the following: (i) an existence or sequence of specified method invocations, (ii) an existence or sequence of specified object creations and deletions, (iii) an existence or sequence of specified class loading and unloading, (iv) values of specified arguments to invocations of specified methods; (v) values of specified return values from invocations of specified methods, and (v) values of specified field values for invoked objects or field values for passed arguments.

28. (Original) The method of claim 24, further comprising the step of collecting said details for at least one of at least one specified number of task instances and at least one specified number of threads.

29. (Currently Amended) The method of claim 24, further comprising the step of dynamically modifying a specification associated with said program task ~~specification~~ associated with said analysis in an iterative process.

30. (Currently Amended) The method of claim 24, further comprising the step of dynamically modifying a said specification ~~of~~ to identify which details to collect in an iterative process.

31. (Original) The method of claim 24, further comprising the step of connecting to a running version of said software system.

32. (Currently Amended) A system for analyzing behavior of a software system, comprising:

a memory that stores computer-readable code; and

5 to implement said computer-readable code, said computer-readable code configured to:

collect details associated with a program task associated with said software system based on a specification associated with said program task, wherein said specification contains one or more conditions to initiate a trace of said program task; and

provide said collected details for analysis.

10

33. (Currently Amended) A system for tracing details associated with a program task executing in a software system, comprising:

a memory that stores computer-readable code; and

15 to implement said computer-readable code, said computer-readable code configured to:

monitor said software system to identify said program task based on a specification associated with said program task, wherein said specification contains one or more conditions to initiate a trace of said program task;and

provide trace details associated with said program task.

20

34. (Currently Amended) An article of manufacture for analyzing behavior of a software system, comprising:

a computer readable medium having computer readable code means embodied thereon, said computer readable program code means comprising:

25 a step to collect details associated with a program task associated with said software system based on a specification associated with said program task, wherein said specification contains one or more conditions to initiate a trace of said program task; and

a step to provide said collected details for analysis.

30 35. (Currently Amended) An article of manufacture for tracing details associated with a program task executing in a software system, comprising:

a computer readable medium having computer readable code means embodied thereon, said computer readable program code means comprising:

a step to monitor said software system to identify said program task based on a specification associated with said program task, wherein said specification contains

5 one or more conditions to initiate a trace of said program task; and

a step to provide trace details associated with said program task.